Enhanced Feature Analysis Framework for Comparative Analysis & Evaluation of Agent Oriented Methodologies

Omankwu, Obinnaya Chinecherem, Nwagu, Chikezie Kenneth, and Inyiama, Hycient

**[1] Computer Science Department, Michael Okpara University of Agriculture, Umudike
Umuahia, Abia State, Nigeria**
saintbeloved@yahoo.com

**[2] Computer Science Department, Nnamdi Azikiwe University, Awka Anambra State, Nigeria,
Nwaguchikeziekenneth@hotmail.com**

**[3] Electronics & Computer Engineering Department, Nnamdi Azikiwe University, Awka
Anambra State, Nigeria.**

ABSTRACT

The objective of this paper is to provide an insight preview into various agent oriented methodologies by using an enhanced comparison framework based on criteria like process related criteria, steps and techniques related criteria, steps and usability criteria, model related or "concepts" related criteria, comparison regarding model related criteria and comparison regarding supportive related criteria. The result also constitutes inputs collected from the users of the agent oriented methodologies through a questionnaire based survey.

Keywords— Agents, Agent Oriented Methodology, Feature Analysis Framework, GAIA, PROMETHEUS, MESSAGE,

1. INTRODUCTION

The objective of this paper is to provide an insight preview into existing agent- oriented methodologies (AOM).Various agent oriented methodologies like GAIA, TROPOS, MAS-COMMONAKADS, PROMETHEUS, PASSI, ADELFE, MASE, RAP, MESSAGE and INGENIAS etc are available and are widely discussed. A comparison of five major agent oriented methodologies: GAIA, TROPS, PROMETHEUS, MESSAGE and MASE are presented in this paper. There had been various types of comparisons [1] done previously also by many researchers and software engineers, these comparisons are based upon certain different criteria [2] like process related criteria, steps and techniques related criteria, steps and usability criteria, model related or "concepts" related criteria, comparison regarding model related criteria and comparison regarding supportive related criteria. All these different comparisons cover almost all features of these methodologies like Application development life cycle support, coverage of life cycle, development approach, type of application domain, agent nature, ease of understanding of development steps etc. Ironically, the "best" methodology cannot be judged as these methodologies are application oriented and none of them can be considered as a perfect template or generalized framework for all kind of agent based applications. The careful evaluation of these methodologies can help developers in choosing the best methodology as per their application requirement.

THE COMPARISON FRAMEWORK

We adopted the feature analysis framework proposed by Tran, Low and Williams [3] as the basis as shown in the figure 1. The feature analysis framework constitutes four criteria: Model related criteria, Technique related criteria, Process related criteria and Supportive Features related criteria.

This framework is capable of assessing AOM (Agent Oriented methodologies) from both the dimensions of conventional system development methodologies and specific to AOSE. Also this framework is also capable of assessing the AOM at a multi-stage level. We are not using the full feature analysis as such but a modified version of the same has been used.

THE EXTENDED COMPARISON FRAMEWORK

The motive of extending the framework is to constitute the classical generic software engineering features [4] [5] [6] in addition to the elements specific to AOSE [7] [8] [9]. Moreover few features of object oriented software engineering are also compared in the framework. In our framework, we are using combination of major attributes of the all the criteria available under the feature analysis framework with the objective to compare the available features in the agent methodologies. These criteria constitute the respective attributes and features along with their description. It will help us assessing the methodologies on some specific guidelines. The details are as under.

A. Model Related Criteria: The model related criteria examine the capabilities and characteristics of the methodology's models and notational components [10]. It also constitutes the concepts represented by the model [11] which is also a basis of our comparison framework. The concept property is divided into three further sub-sections: Internal properties, social properties and technical properties. It constitutes (a) AUTONOMY, which states that Agents can execute, operate and can be self-decisive of their own without any direct/external human intervention. Agents must have an inherent control on their internal state which is dynamic in nature and can be modified by taking inputs from other agents in the environment. (b) REACTIVITY, which states that agents should respond in a consistent way towards changes occurring in the environment. The changes are triggered by the other agents present in the environment. (c) CONCURRENCY, which states that agents must interact with other agents simultaneously to achieve more than one goal. (d) PRO-ACTIVENESS, which states that agents must keep track of their goals evolving over time. Goals can evolve due to the changes in the environment. (e) ENVIRONMENT BELIEF, which states that agents must receive inputs from the environment, act accordingly and then may provide output to the environment, which can be used by other agents working in the environment. (f) COOPERATIVE BEHAVIOR, which states that Agents can request, respond, deny and even negotiate with other agents [12] in order to perform their individual goals and the system goals. (g) COMMUNICATION ABILITY, which states that agents can communicate directly, transitively, single directional (one to one) or multi-directional like a broadcast system [13]. (h) ACP (Agent Communication Protocol), which state that different agents communicate with each other by the means of message passing [12] [13]. These messages may be two fold also. A valid sequence of messages is required in order to achieve the goal(s). (i) ACL (Agent Communication Language), ACL provides agents with a mean of exchanging information and knowledge between them [13]. Using ACL, agents transport messages over the network using low level and high level protocols. (j) COMPLETENESS & EXPRESSIVENESS, to model the system from architectural view point as well as from the unit view point. (k) CONSISTENCY, This property requires that there should be no contradiction between models [4]. (l) MODEL REUSABILITY, the ability of any component to be re-used by other system with minor or even no modifications. (m) ABSTRACTION & MODULARITY, abstraction deals with the ability of the AOM to produce models at various levels of details

Modularity is the property to divide the system in small manageable chunks.

Technique Related Criteria: This criteria deal with assessing the methodology's techniques to perform development steps and/or to produce models and notational components.

(a) AVAILABILITY OF TECHNIQUES & HEURISTICS, This is the property of an AOM to provide techniques to perform each process step. Techniques to produce each model and notational components.
(b) TECHNIQUE USABILITY, AOM should provide a systematic structure to be followed in order to develop a system model.
(c) EASE OF UNDERSTANDING, The notations provided by the AOM must be easy to learn & remember by different type of users [14]. This requires inclusion of the symbols and notations which are familiar to the users.

Process Related Criteria: This criteria looks at the applicability of the AOM, the steps provided for development process and the development approach followed by the AOM.

(a) DEVELOPMENT LIFE CYCLE, This criteria state about the development context supported by the AOM. Whether it supports waterfall model, prototype model, iterative enhancement model etc.

(b) DEVELOPMENT PROCESS STEPS, This criterion evaluates the tasks and activities specified by the AOM for the development process.

(c) VERIFICATION & VALIDATION SUPPORT, Are we building the system right? Are we building the right system? Both the questions must be answered in order to have a clear idea about correctness of the developed models and specified requirements.

(d) REFINABILITY, a simplified sequence of steps must be provided by the methodology to add new details in the existing model. Refinement allows the developers to make necessary changes at gradual stages of design development in an easy and simplified way [15].

Support Features Related Criteria: These are "add on" features provided by any methodology. This criterion assesses various supplementary features provided by any AOM. It includes CASE tools to support dynamic and open systems which allow dynamic addition and removal of the agents. Support for mobile agents and conjunction of conventional objects in the MAS are also included in the supportive features.

SOFTWARE & METHODOLOGICAL SUPPORT, This criterion assesses availability of various development support tools like CASE tools and libraries to develop MAS. (b) OPEN SYSTEM DEVELOPMENT SUPPORT, Multi-agent systems are dynamic in nature. Various agents interact with each other to perform their goal. In a dynamic open system, agents can be added or removed in and from the system at any point of time. This criterion assesses support provided by an AOM to develop open agent based system.

EVALUATION RESULTS

We have compared features of five AOMs GAIA, MaSE, PROMETHEUS, TROPOS & MESSAGE using the above mentioned feature analysis framework. We have done a primary survey also using a questionnaire among the users of these methodologies along with presenting the claims made by the developer of particular AOMs and we have also included our experience regarding the same. The primary research work is done to sideline the chances of biasing towards a particular AOM. The questionnaire consists of twenty one questions in total divided into four sections and is based upon the modified comparison framework discussed in earlier section. The Combined result of the questionnaire and our observation is presented in the subsequent sections. The results are mentioned on an abstract scale of H, M, L, N and X where H stands for High, M for Medium, L for Low, N for Not Available, X for can't say. The results are as under.

AUTONOMY: From the comparison provided in table 1, we can analyze that almost all the five AOMs are having high ratings with respect to this property. This due to the fact that all of these AOMs have constructs available to implement the autonomous property. For instance TROPOS has plan diagrams, PROMETHEUS has plan descriptor and MASE consist of task state diagram. These constructs are good enough to implement agent plans and reasoning rules in order to implement the autonomous property for an agent based system. **REACTIVITY:** As shown in the comparison table 1, again the rating is on high to medium scale. TROPOS has the Actor Diagrams and PROMETHEUS has the Agent Class Descriptor to implement the reactive behavior of the agent based system**.** **CONCURRENCY:** PROMETHEUS and GAIA are considered to be having very low facility available for concurrency; the strongest AOM for concurrency implementation is MaSE due to availability of constructs like Task State Diagrams & Communication Class Diagram. These diagrams are helpful to define coordination protocols between two agents and thus achieving the important attribute of Concurrency. **PRO-ACTIVENESS:** Except MESSAGE, we got approximately similar response for all five AOMs regarding this criterion. PROMETHEUS seems to be the best amongst all due to availability of Action Descriptors. Using this construct the agents can be modeled in the way such that they respond to the goals evolving over time in the environment. **ENVIRONMENT BELIEF:** From the survey and our own observations, we found that GAIA and PROMETHEUS provide clear constructs for Environment Belief.

They allow modeling of agents in the way such that agents can capture information from the environment and appropriate processing can be done.

GAIA provides Environmental Model and PROMETHEUS provides System Overview Diagram which is also known as the Environmental Model. **COOPERATIVE BEHAVIOR:** Agents cannot exist in a vacuum. Agents need support of other agents and they have to provide support also. Agents can delegate their task to other agents, can negotiate with other agents and can work in a shared way also. Acquaintance of one agent with other can be easily modeled in GAIA as they provide acquaintance model for the same,

TROPOS has Sequence/Collaboration diagram, PROMETHEUS provides Interaction Diagram, MaSE provides Agent Class diagram and MESSAGE provides Organization Model. From our own experience, we found that the constructs provided by TROPOS & MESSAGE are capable of modeling any kind of agent acquaintance.

COMMUNICATION ABILITY: Agents can directly, indirectly, synchronously and asynchronously interact with each other. From the survey and our own experience we felt that almost all five AOMs provide satisfactory constructs for the variety of communication modes. **ACP (AGENT COMMUNICATION PROTOCOL):** All five AOMs in consideration provide good constructs with some minor limitations. GAIA provides Interaction model for the same but with the limitation that contents of exchanged message between agents cannot be defined in GAIA model. TROPOS provides Sequence Diagrams, PROMETHEUS provides Interaction Protocol Diagrams, MaSE provides Communication Class Diagram and MESSAGE provides Interaction Model.

ACL (AGENT COMMUNICATION LANGUAGE): An ACL provides means to agents to exchange information and knowledge with other agents. All five AOM's basic communication language is based upon the *"speech act"* where not only contents but intentions and actions also matters. Both KQML and FIPA-ACL are supported by all the AOMs under consideration. In table 1, from the survey and from our observation, it can be seen that the ratings are medium to high for all the AOMs. **COMPLETENESS & EXPRESSIVENESS:** From the survey and from our own experience we analyzed that all five AOM provide fairly good & well defined symbols and notations. GAIA, MaSE, TROPOS, MESSAGE and PROMETHEUS provide adequate constructs to completely express complex and dynamic system. One of the team members does not found syntax and symbols of GAIA and TROPOS satisfactory to completely model MAS, but from our own experience we felt that the TROPOS may be the possible candidate for the claim made by the team, as it only provides some help during detailed design. For GAIA, we felt that symbols and notations are quite satisfactory. **CONSISTENCY:** This attribute differs a lot from one AOM to another. Consistency requires there must be a consistent relationship between modeling and design i.e. inter-model and intra-model consistency. The Prometheus Design tool (PDT) in PROMETHEUS and agent-Tool in MaSE provide enough support for the design and model consistency check.

MESSAGE only provide limited consistency check in the form drawing diagrams. GAIA and TROPOS do not provide support for consistency check at all.

MODEL REUSABILITY: Either the teams are not sure about availability of this criterion or they felt it at very low level. This is due to the fact that none of the AOM under consideration provides any explicit construct to implement model reusability. Though TROPOS, PROMETHEUS and MaSE claims for model reusability bur no formal guidelines are available to design reusable components in any of the AOM.

ABSTRACTION & MODULARITY: Almost all the five AOMs are having medium to high ratings in this criteria. The Agent model in GAIA, Agent/Role model in MESSAGE and Agent Class Diagram in MaSE etc. provides sufficient constructs for achieving abstraction and modularity while modeling or designing any MAS.

Technique Related Criteria constitutes:

AVAILABILITY OF TECHNIQUES & HEURISTICS: It deals with the availability of clearly defined techniques to perform each process step and to produce each model and notational components. It is observed that all five AOMs under consideration provide fairly good support at each step either implicitly or explicitly. Right from identifying system tasks to system deployment, all AOMs provide various constructs for the same. For instance GAIA provides Role Model for Identifying system tasks in an implicit way and Agent Model for specifying agent classes.

Other techniques provided by GAIA are Interaction Model, Service Model, and Environmental Model etc. being used at different steps. TROPOS has Actor Diagram, Plan Diagram and Sequence Diagram. PROMETHEUS has Goal Diagram, Agent Class Descriptor, Interaction Diagrams & Protocols and Capability Diagram etc. MaSE provides Goal Hierarchy Diagram, Agent Class Diagram, Communication Class Diagram and Deployment Diagram etc. MESSAGE has Task Model, System Architecture Diagram and Organizational Model etc.

TECHNIQUE USABILITY: As discussed earlier, MaSE, PROMETHEUS & MESSAGE provides have integrated tool support to draw diagrams and check model & design consistency. TROPOS is an exception where no such kind of facility is available, though the developers claim that the notations and symbols are fairly easy to understand. Still, we have maintained low ratings for the TROPOS regarding usability criteria.

EASE OF UNDERSTANDING: "*Ease of learning*" any AOM is concerned with many criteria like unambiguous syntax and semantics, clear and expressiveness nature etc. All the AOMs under consideration are having medium to high ratings for this criteria.

DEVELOPMENT LIFE CYCLE: It deals with the nature of development life cycle. GAIA supports iterative development within each phase but sequential between phases, TROPOS is iterative & incremental, PROMETHEUS & MaSE are iterative across all phases and MESSAGE follows RUP life cycle. So rather than giving the ratings to a particular AOM, we have mentioned the nature of development life cycle being followed by the AOM.

DEVELOPMENT PROCESS STEPS: It deals with the clear separation of phases/steps of development process. We observed that except MaSE, no AOM provides clear and explicit steps for Testing, Debugging, Deployment and Maintenance. Even Implementation step is also not provided at satisfactory level by all of the AOMs. All the AOMs though provide good support for Requirement Analysis, Architectural Design and Detailed Design. Considerable work still needs to be done in the later phases the development process of agent oriented system**.**

VERIFICATION & VALIDATION SUPPORT: V & V is an essential activity to achieve quality of an agent based system. Except MaSE and PROMETHEUS, no AOM under consideration provides support for verification and validation. MESSAGE has kept this feature as future development.

REFINABILITY: Again, all the AOMs got medium to high ratings for this feature. Developers are free to roam in different process steps to enhance the details in the existing model.

CONCLUSION & CRITICAL DISCUSSION

There is very strong demand of developing complex software systems for industrial and general applications. Agents seem to be the best solution for the same. Attributes like autonomy, reactivity, pro-activeness etc provide a fantastic platform to design and develop complex systems. The need is to compare and evaluate various advantages and disadvantages of various AOMs. As various AOMs has been proposed and discussed in the literature, the aim of this paper is to provide an unbiased comparison of different methodologies using a modified feature analysis framework along with the primary survey technique. We have carried out comparison of five selected AOMs. The purpose of the comparative study is not to prove one AOM superior or inferior over another but to figure out strengths, weakness, domain applicability, similarities and dissimilarities as compared to each other.

The aim of modifying the feature analysis framework is to compare the software engineering attributes provided by a particular AOM in terms of classical software engineering paradigms, object oriented paradigms and those which are specific to agent based development.

Another reason is to provide a multistage comparative analysis to cover all significant software engineering criteria. This is required to enhance the overall software development experience of the developing team to develop a complex system using agent oriented software engineering paradigms. We have incorporated four basic paradigms which an AOM supports. These are model related criteria, technique related criteria, process related criteria and supportive features related criteria. In addition to the qualitative analysis of the AOMs we have also used the primary research technique of questionnaire, where we have collected views of the users of agent oriented methodologies. This is required to eliminate the chances of any kind of biasing in the comparative analysis.

From the results obtained we can see that all five AOMs provide reasonably good support for the features like pro-activeness, autonomy, reactivity etc. required for developing an agent based application. All five AOMs are also considered as pure agent oriented methodologies rather than merely the extension of object oriented methodologies. All five AOMs have clear and understandable notations to model and develop the agent based system. Along with some good similarities the AOMs under consideration have some dissimilarity also. TROPOS seems to be difficult to use and understand. MaSE

GAIA seems to be providing less support in terms of expressiveness. Only PROMETHEUS & MaSE provides tool support to check consistency between models. MESSAGE and GAIA does not provide support for detailed design. On one hand PROMETHEUS & MaSE provide good heuristics support for architectural and detailed design, on the other hand MESSAGE provide no heuristic support for the same. In addition to the individual pros and cons, all AOMs share some good and bad points.

None of the five AOMs under consideration provide explicit feature to design team work in multi agent system. Environmental modeling constructs are also not fully provided in the all five AOMs. Implementation, Testing, Debugging and maintenance phases are either poorly defined or not defined at all in all five AOMs.

Various other factors which are important in industrial terms like project management techniques, software quality assurance techniques, cost and effort estimation etc. are also not included in any of the AOM. An AOM which can be considered as a benchmark for developing an agent oriented system with all relevant feature support is still to be developed; the need is to the use the positive features of each AOM and develops a perfect AOM rather then moving in separate and scattered directions.

References

[1] David Law, "*Methods for Comparing Methods: Techniques in Software Development*". NCC Publications, 1988.

[2] Barbara Kitchenham, "*DESMET:a method for evaluating software engineering methods and tools*." Technical Report TR96-09, University of Keele, U.K., August 1996.

[3] Tran, Q.N., Low, G., Williams, M.A.," *A Feature Analysis Framework for Evaluating Multiagent System Development Methodologies.*" In Zhong, N., Ras, Z.W., Tsumoto, S., Suzuki, E. (eds): Foundations of Intelligent Systems – Proc. of the 14th Int. Symposium on Methodologies for Intelligent Systems ISMIS'03 (2003) pp 613-617.

[4] Wood B., Pethia R., Gold L.R., and Firth R."*A guide to the assessment of software development methods. Technical Report 88-TR-8*", Software Engineering Institute, Carnegie-Mellon University, Pittsburgh, PA, 1988.

[5] Jayaratna N.," *Understanding and evaluating methodologies – NIMSAD a systematic framework"*. Maidenhead, UK: McGraw-Hill. 1994

[6] Olle T.W., Sol H.G., & Tully, C.J., "*Information systems design methodologies – A feature analysis*." Amsterdam: Elsevier Science Publishers. 1983

[7] O'Malley S.A. & DeLoach S.A."*Determining when to use an agent oriented software engineering paradigm*". In *Proceedings of the 2nd International Workshop on Agent-Oriented Software Engineering (AOSE 2001)*, Montreal, Canada, (pp. 188-205). Springer-Verlag. May 29, 2001.

[8] Cernuzzi L. & Rossi G.,"*Evaluation of agent-oriented modeling methods*". In *Proceedings of the OOPSLA Workshop on Agent-Oriented Methodologies*, Seattle, (pp. 21-33). University of Technology, Sydney: Centre for Object Technology Applications and Research. November 4-8 , 2002

[9] Sabas A., Badri M., & Delisle, S.,"*A multidimensional framework for the evaluation of multi-agent system methodologies*". In *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, (pp. 211-216). Orlando: International Institute of Informatics and Systemics. July 14-18, 2002

[10] Frank U."*Evaluating modeling languages:relevant issues, epistemological challenges and a preliminary research framework. Technical Report 15*", Arbetsberichte des Instituts fuer Wirtshaftsinformatik (Universitt Koblenz-Landau), 1998.

[11] Avison D. and Fitzgerald G., "*Information Systems Development: Methodologies, Techniques and Tools*." McGraw-Hill, New York, 2nd edition, 1995, pp 452.

[12] Jacques Ferber.,"*Multi-agent Systems: An Introduction to Distributed Artificial Intelligence."* Addison-Wesley, 1999 pp 78-82.

[13] Munindar P. Singh, "*Agent communication languages: Re-thinking the principles*." IEEE Computer, 31(12):40–47, December 1998.

[14] Rumbaugh J. "*Notation notes: Principles for choosing notation*." Journal of Object-Oriented Programming (JOOP), 8(10):11–14, May 1996.

[15] Nicholas R. Jennings."*An agent-based approach for building complex software systems*." Communications of the ACM, 44(4):35–41, 2001.